International Journal of Engineering Research (ISSN : 2319-6890)
Volume No.2, Issue No.3, pp : 234-238 01 July 2013

Performance Comparison of Turbo Code in WIMAX System with Various Detection Techniques

Vikas Tursenia[1]
Department of Electronics and Communication Engineering, IET-DAVV, Indore, (MP)- INDIA
vikas.t12i@gmail.com

Abstract

The different FEC techniques like convolution code, RS code and turbo code are used to improve the performance of communication system. In this paper, we study the performance of the MAP, Log-MAP, Max-Log-MAP and APP decoding algorithms for turbo codes, in terms of the a priori information, a posteriori information, extrinsic information and channel reliability. We also analyze how important an accurate estimate of channel reliability factor is to the good performances of the iterative turbo decoder. The simulations are made for parallel concatenation of two recursive systematic convolution codes with a block interleaver at the transmitter, AWGN channel and iterative decoding with different algorithms at the receiver side. The comparison of these detection techniques in term of BER performance is discussed in result section.

Keywords-component; MAP decoding, Log-MAP decoding, MAX-Log-MAP decoding, Baye's Theorem, APP decoding.

I. Introduction

An ever crowed Radio spectrum system implies that future demands must be met using more data throughput wireless technologies. Since system bandwidth is limited and user demand continues to grow. This problem could be solved by WiMAX (Worldwide Interoperability for Microwave Access) technology based on the IEEE 802.16 specifications. Since it offers high level of services. In mobile wireless environment the channel is hostile and behaves adversely. To transmit the faithful data over these system the BER performance is further improved using forward error correction codes (FEC). Convolution code ,Reed Solomn and Turbo Codes are considered for this purpose. Since FEC codes require only simplex communication, they are especially attractive in wireless communication systems, helping to improve the energy efficiency of the system .

II. Turbo code

In 1993 Berrou, Glavieux and Thitimajshima2 proposed "a new class of convolution codes called turbo codes whose performance in terms of Bit Error Rate (BER) are close to the Shannon limit".

It is theoretically possible to approach the Shannon limit by using a block code with large block length or a convolutional code with a large constraint length. The processing power required to decode such long codes makes this approach impractical. Turbo codes overcome this limitation by using recursive coders and iterative decoders. The recursive coder makes convolutional codes with short constraint length appear to be block codes with a large block length, and the iterative soft decoder progressively improves the estimate of the received message. In this paper we mainly focus on three different technique of decoding of turbo code which is APP, MAX and MAX*.

In a typical turbo decoding system (see Fig. 1), two decoders operate iteratively and pass their decisions to each other after each iteration. These decoders should produce soft-outputs to improve the decoding performance. Such a decoder is called a Soft-Input Soft- Output (SISO) decoder [9]. Each decoder operates not only its own input but also on the other decoder's incompletely decoded output which resembles the operation principle of turbo engines. This analogy between the operation of the turbo decoder and the turbo engine gives this coding technique its name, "turbo codes". The decoding process of turbo code is shown in figure 1.

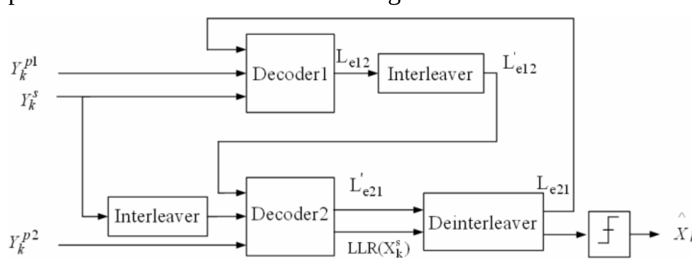

Figure.1 Block diagram of turbo decoder

Turbo decoding process can be explained as follows: Encoded information sequence X_k is transmitted over an Additive White Gaussian Noise (AWGN) channel, and a noisy received sequence Y_k is obtained. Each decoder calculates the Log-Likelihood Ratio (LLR) for the k^{th} data bit d_k, as

$$L(d_k)=\log\left[\frac{P(d_k=1|Y))}{P(d_k=0|Y)}\right] \qquad (1)$$

International Journal of Engineering Research (ISSN : 2319-6890)
Volume No.2, Issue No.3, pp : 234-238 01 July 2013

LLR can be decomposed into 3 independent terms, as

$$L(d_k)=L_{apri}(d_k)+Lc(d_k)+Le(d_k) \quad (2)$$

Where $L_{apri}(d_k)$ is the a-priori information of d_k, $L_c(x_k)$ is the channel measurement, and $L_e(d_k)$ is the extrinsic information exchanged between the constituent decoders. Extrinsic information from one decoder becomes the a-priori information for the other decoder at the next decoding stage.
L_{e12} and L_{e21} in Figure 1 represent the extrinsic information from decoder1to decoder2 and decoder2 to decoder1 respectively.
LLR computations can be performed by following method.

A. The MAP Algorithm

The MAP algorithm is an optimal but computationally complex SISO algorithm. The Log-MAP and Max-Log-MAP algorithms are simplified versions of the MAP algorithm.

MAP algorithm calculates LLRs for each information bit as

$$L(dk)=\ln\left[\frac{\sum_{s_k}\sum_{s_{k-1}} \Upsilon_1(S_{k-1}.s_k)\alpha(s_{k-1})\beta(s_k)}{\sum_{s_k}\sum_{s_{k-1}} \Upsilon_0(S_{k-1}.s_k)\alpha(s_{k-1})\beta(s_k)}\right] \quad (3)$$

where α is the forward state metric, β is the backward state metric,γ is the branch metric, and S_k is the trellis state at trellis time k . Forward state metrics are calculated by a forward recursion from trellis time k = 1 to, k = N where N is the number of information bits in one data frame. Recursive calculation of forward state metrics is performed as

$$\alpha_k(s_k) = \sum_{j-1}^{1} \alpha_{k-1}(S_{k-1})\Upsilon_j(S_{k-1.}S_k) \quad (4)$$

Similarly, the backward state metrics are calculated by a backward recursion from trellis time k = N to, k = 1 as

$$\beta_k(s_k) = \sum_{j-1}^{1} \beta_{k-1}(S_{k-1})\Upsilon_j(S_{k.}S_{k+1}) \quad (5)$$

Branch metrics are calculated for each possible trellis transition as

$$\Upsilon_i(S_{k-1,}S_k) = A_k P(S_k|S_{k-1}) \exp\left[\frac{2}{N_o}\left(y_k^5 x_k^5(i) + \right.\right.$$
$$ykpxkpi,Sk-1,sk \quad (6)$$

Where i = (0,1) , A_k is a constant, x_k^5 and x_k^p are the encoded systematic data bit and parity bit, and, $y_k^5$and y_k^p are the received noisy systematic data bit and parity bit respectively.

B. The Log-MAP Algorithm

To avoid complex mathematical calculations of MAP decoding, computations can be performed in the logarithmic domain. Furthermore, logarithm and exponential computations can be eliminated by the following approximation

$$\text{Max*}(x,y) \triangleq \ln(e^x + e^y) = \max(x,y)+\log(1+e^{-(y-x)}) \quad (7)$$

The last term in max*(.) operation can easily be calculated by using a look-up table (LUT).

So equations (3)-(6) become

$$L(d_k)=max^*_{s_{k-1},s_k,1}(\Upsilon(S_{k-1},S_k)+\alpha_{k-1}(S_{k-1})+\beta_k(S_k)) \quad (8)$$

$$max^*_{s_{k-1},s_k,0}(\Upsilon(S_{k-1},S_k)+\alpha_{k-1}(S_{k-1})+\beta_k(S_k)) \quad (9)$$

$$\alpha_k(S_k) = max^*_{s_{k-1},i}(\alpha_{k-1}(S_{k-1}) + \Upsilon_i(s_{k-1},s_k)) \quad (10)$$

$$\beta_k(S_k) = max^*_{s_k,i}(\alpha_{k-1}(S_{k+1}) + \Upsilon_i(s_k,s_{k+1})) \quad (11)$$

$$\Upsilon_i(s_{k-1}s_k) = \frac{2}{N_o}(y_k^s x_k^s(i) + y_k^p x_k^p(i,s_{k-1},s_k) + \ln(P(S_k|S_{k-1}))+K \quad (12)$$

Where K is a constant.

C. The Max-Log-MAP Algorithm

The correction function $f_c = \log(1+ e^{-|y-x|})$ in the max*(.) operation can be implemented in different ways. The Max-log-MAP algorithm simply neglects the correction term and approximates the max* (.) operator as.

$$\ln(e^x + e^y) \approx \max(x,y) \quad (13)$$

at the expense of some performance degradation.
This simplification eliminates the need for an LUT required to find the corresponding correction factor in the * max (.) operation. The performance degradation due to this simplification is about 0.5dB compared to the Log- MAP algorithm.

D. Baye's Theorem of conditional probabilities

Let's first state the theorem of conditional probability, also called the Baye's theorem.

$$P\ (A \text{ and } B)=P(A)P(B \text{ given } A) \quad (14)$$

Which we can write in more formal terminology as

$$P\ (A,B)=P(A)P(B|A) \quad (15)$$

International Journal of Engineering Research (ISSN : 2319-6890)
Volume No.2, Issue No.3, pp : 234-238 01 July 2013

Where $P(B|A)$ is referred to as the probability of event B given that A has already occurred. If event A always occurs with event B, then we can write the following expression for the absolute probability of event A.

$$P(A)=\sum_B P(A,B) \quad (16)$$

If events A and B are independent from each other then (A) degenerates to

$$P(A,B)=P(A) \quad (17)$$

$$P(A,B)=P(A)P(B) \quad (18)$$

The relationship C is very important and we will use it heavily in the explanation of Turbo decoding in this chapter

E. APP Decoding

A-priori and a-posteriori probabilities

Here is Bayes' theorem again.

$$P(A,B)=P(AB)\ P(B) \quad (19)$$

P(A,B)=	Probability of both A and B
P(AB) =	a posteriori probability of event A
P(B) =	a priori probability of event B

The probability of event A conditioned on event B, is given by the probability of A given B times the probability of event a.

The probability of A, or P(A) is the base probability of even A and is called the a-priori probability. The term P(A,B) the conditional probability is called the a-posteriori probability or APP. One is independent probability, the other depends on some event occurring. We will be using the acronym APP a lot, so make sure you remember that is the same a-posteriori probability. In other words, the APP of an event is a function of an event also occurring at the same time. We can write as

$$P(A,B)=P(AB)=\frac{P(B|A)P(A)}{P(B)} \quad (20)$$

$$P(AB)=APP$$

The logarithm of likelihood ratio (LLR), $L(d_k)$ associated with each decoded bit d_k

$$L(d_k)=\log\left[\frac{P(d_k=1|Y))}{P(d_k=0|Y)}\right] \quad (21)$$

Where $P(d_k=i|y)$, i=0,1 is the a posteriori probability (APP) of the data bit d_k.

The APP of a decoded data bit d_k can be derived from the joint probability λ_k^i(m) defined by

$$\lambda_k^i = P_r\{d_k=i, S_k=m/R_1^N\} \quad (22)$$

And thus, the APP of a decoded data bit d_k is equal to

$$P_r\{d_k=i/R_1^N\}=\sum_M \lambda_k^i\ (m),\ i=0,1 \quad (23)$$

From equation (21)

$$L(d_k)=\log\left[\frac{\sum_m \lambda_k^1(m)}{\sum_m \lambda_k^0(m)}\right] \quad (24)$$

Finally the decoder can make a decision by comparing $L(d_k)$ to a threshold equal to zero

$$d_k=1 \quad \text{if} \quad L(dk)>0$$
$$d_k=1 \quad \text{if} \quad L(dk)<0$$

in order to compute the probability $\lambda_k^1(m)$ let us introduce the probability function $\alpha_k^1(m)$, $\beta_k^1(m)$ and $\Upsilon_i(R_k, m',$m)

$$\alpha_k^1(m) = \frac{\Pr\{\mathrm{dk} = \mathrm{i}, \mathrm{S_k} = \mathrm{m}, R_1^k\}}{P_r\{R_l^k\}} P_r\{\mathrm{dk} = \mathrm{i}, \mathrm{S_k} = \mathrm{m}, R_1^k\} \quad (25)$$

$$\beta_k^1(m) = \frac{\Pr\{\frac{R_{k+1}^N}{S_k}=m\}}{P_r\{R_{k+1}^N/R_1^k\}} \quad (26)$$

$$\Upsilon_i(R_k, m', m) = \left\{\mathrm{d_k} = \mathrm{i}, \mathrm{R_k}, \mathrm{S_k} = \frac{\mathrm{m}}{\mathrm{S_{k-1}}} = \mathrm{m'}\right\} \quad (27)$$

The joint probability $\lambda_k^1(m)$ can be rewritten using baye's rule

$$\lambda_k^i(m) = \frac{\Pr\{d_k=i,S_k=m,R_1^k,R_{k+1}^N\}}{P_r\{R_l^k,R_{k+1}^N\}} \quad (28)$$

Thus we obtain

$$\lambda_k^i(m) = \frac{\Pr\{\mathrm{dk} = \mathrm{i}, \mathrm{S_k} = \mathrm{m}, R_1^k\}}{P_r\{R_l^k\}} \frac{\Pr\{\mathrm{d_k} = \mathrm{i}, \mathrm{S_k} = \mathrm{m}, R_1^k, R_{k+1}^N\}}{P_r\{R_l^k/R_{k+1}^N\}} \quad (29)$$

Taking into account the event after time k are not influenced by observation R_l^k and bit d_k if state S_k is known, the $\lambda_k^i(m)$ is equal.

$$\lambda_k^i(m)=\alpha_k^1(m)\beta_k^1(m) \quad (30)$$

The probabilities $\alpha_k^1(m)$ and $\beta_k^1(m)$ can be recursively calculated from probability $\Upsilon_i(R_k, m', m)$ .From annex I, we obtain

International Journal of Engineering Research (ISSN : 2319-6890)
Volume No.2, Issue No.3, pp : 234-238 01 July 2013

$$\alpha_k^1(m) = \frac{\sum_{m'} \sum_{j=0}^{l} \Upsilon_i(R_k, m', m)\alpha_{k+1}^j(m')}{\sum_m \sum_{m'} \sum_{m'} \sum_{j=0}^{l} \Upsilon_i(R_k, m', m)\alpha_{k+1}^j(m')} \quad (31)$$

And

$$\beta_k^1(m) = \frac{\sum_{m'} \sum_{j=0}^{l} \Upsilon_i(R_{k+1}, m', m)\beta_{k+1}^j(m')}{\sum_m \sum_{m'} \sum_{m'} \sum_{j=0}^{l} \Upsilon_i(R_{k+1}, m', m)\alpha_{k+1}^j(m')} \quad (32)$$

III. RESULT

The WiMax system are simulated with the following parameters

S. No.	Parameter	Specification
1	FFT size	256
2	Cyclic Prefix	1/4
3	FEC code	Turbo coding
4	FEC decoding	True APP , MAX , MAX*
5	Channel	AWGN
6	Modulation technique	BPSK,QPSK,QAM8
7	SNR range	-10:18

The comparisons have been performed with the different detection technique for different modulation scheme. The AWGN channel is used to avoid the complexity of fading effect and determine good BER performance .The result is given below.

a) BER performance with BPSK modulation.

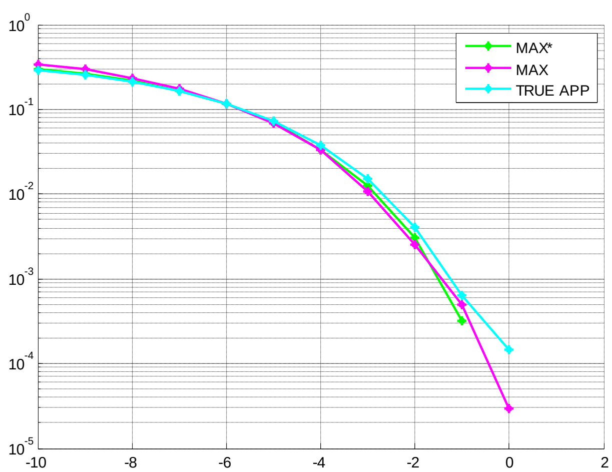

Figure.2 BER at BPSK

Max* has provide slightly better performance as compare to other detection techniques.

b) BER performance with QPSK modulation.

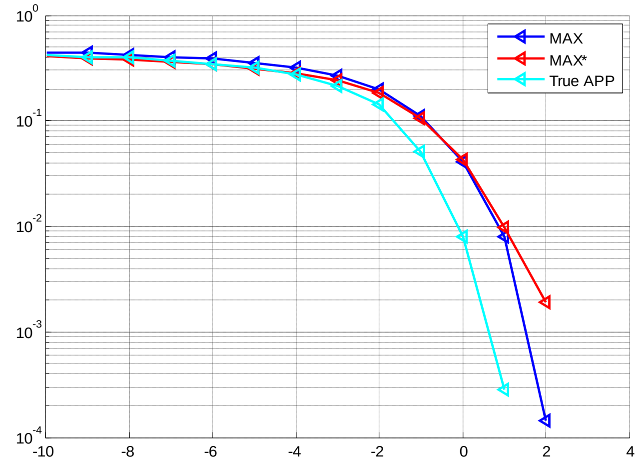

Figure.3 BER at QPSK

True APP at this modulation is performing better.

c) BER performance with QAM8 modulation :

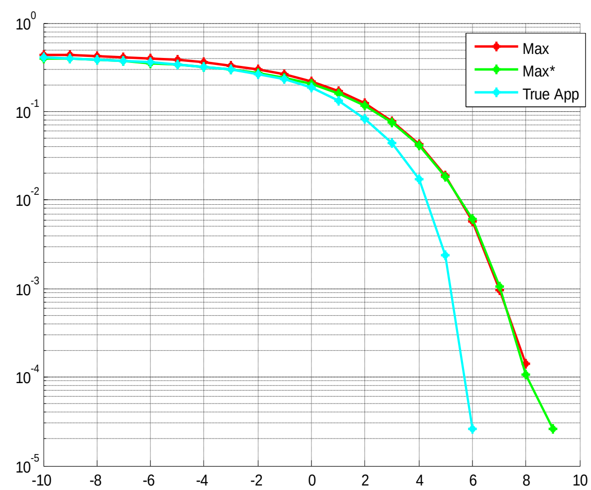

Figure.4 BER at QAM-8

IV. CONCLUSION

The BER performance is evaluated for various detection technique of turbo code. The true APP technique are found better than MAX and MAX* detection techniques. The numbers of iteration are reducing to half in the true APP compare to other for similar BER performance.

V. REFERENCES

i. *maximum a posteriori decoding algorithms for turbo codes by hamid r. sadjadpour, at&t research-shannon labs, florham park, nj,*

ii. *a-posteriori probability decoding of variable-length codes using a three-dimensional trellis representation by j¨org kliewer and ragnar thobaben, 2003, university of kiel, institute for circuits and systems theory, 24143 kiel, germany*

iii. *maximum a posteriori decoding of turbo codes by bernard sklar*

iv. *extrinsic information in iterative decoding: a unified view giulio colavolpe, associate member, ieee, gianluigi ferrari, student member, ieee, and riccardo raheli, member, ieee*

v. *a turbo tutorial by jakob dahl andersen department of telecommunication technical university of denmark*

vi. *a new method for performance evaluation of bit decoding algorithms using statistics of the log likelihood ratio by ali abedia,_, amir k. khandanib*

vii. *a soft-input soft-output maximum a posteriori (map) module to decode parallel and serial concatenated codes s. benedetto,a d. divsalar,b g. montorsi,a and f. pollarab*

viii. *from bcjr to turbo decoding: map algorithms made easier by silvio a. abrantes, april 2004.*

xvii. *a. bartolazzialenia aerispazio*

ix. *maximum a posteriori decoding algorithms for turbo codes by hamid r. sadjadpour at&t research-shannon labs, florham park,nj.*

x. *methods to reduce memory requirements of turbo codes by vahid garousi.*

xi. *the decision reliability of map, log-map, max-log-map and sova algorithms for turbo codes by lucian andrei perişoară, and rodica stoian.*

xii. *turbo codes—a new pccc design spyridon k. chronopoulos, giorgos tatsis, panos kostarakis physics department, university of ioannina, ioannina, greece.*

xiii. *reduced-complexity decoding for for concatenated codes based on rectangular parity-check codes and turbo codes john m. shea and tan f. wong university of florida department of electrical and computer engineering gainesville, florida.*

xiv. *zigzag codes and concatenated zigzag codes li ping, member, ieee, xiaoling huang, and nam phamdo, senior member, ieee ieee transactions on information theory, vol. 47, no. 2, february 2001.*

xv. *noncoherent turbo decoding jacob vainappel, einat hardy, and dan raphaeli.*

xvi. *novel dvb-rcs standard turbo code:details and performances of a decoding algorithm d. giancristofaro,*